

### **AMENDMENTS TO THE CLAIMS**

The following listing of claims will replace all prior versions and listings of claims in the application.

#### **LISTING OF CLAIMS**

1. (Currently Amended) A machine-implemented method for operating a kernel of a computing apparatus comprising:

receiving, from a user of the kernel, proposed changes to a plurality of tunable parameters of tunable kernel modules;

automatically determining whether application of the changes would violate a set of constraints, the determination including automatically evaluating one or more alternate orders by which the proposed changes can be effectuated to identify whether a valid order exists that would not violate the set of constraints; and

effectuating either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively, the effectuation including using the valid order if such is identified by the automatically evaluating step.

2. (Previously Presented) A method for operating a computing apparatus comprising:  
specifying proposed changes to a plurality of tunable parameters of tunable kernel modules;  
determining whether application of the changes would violate a set of constraints;  
effectuating either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively;

booting an operating system kernel having a tunable parameter registration function;  
installing a plurality of kernel modules, wherein at least some of said kernel modules are said

tunable kernel modules having at least one said tunable parameter and at least one handler function, said handler function configured to at least one of obtain or change a value of a registered tunable parameter, wherein installing said tunable kernel modules includes writing information about its at least one tunable parameter into a registry memory maintained by the operating system kernel and indicating that the at least one tunable parameter is available for at least one of read or write access; and

using the tunable parameter registration function and the registry memory, registering the at least one tunable parameters of more than one kernel module;

and further wherein said specifying proposed changes to a plurality of tunable parameters comprises invoking one of said handler functions with a set of proposed new values for a corresponding selected subset of the registered tunable parameters;

wherein said determining whether application of the changes would violate a set of constraints comprises determining whether the proposed new values are valid for every tunable parameter in the selected subset of the registered tunable parameters; and

wherein said effectuating either none or all of the changes comprises changing values of either every one or none of the selected subset of registered tunable parameters to the corresponding new values, conditioned upon a result of said determination of whether the proposed new values are valid.

3. (Original) A method in accordance with Claim 2 wherein one of the tunable kernel modules is a dynamically loadable tunable module (DLKM), and further wherein registering the at least one tunable parameter of the DLKM is performed the first time the DLKM is loaded; said method further comprising:

removing the DLKM from memory;

maintaining the registration of said at least one tunable parameter of the DLKM while the DLKM is no longer in memory;

loading the DLKM into memory at least a second time; and

accessing the value of at least one said tunable parameter of the DLKM using at least one said handler function, after the DLKM is loaded into memory at least a second time.

4. (Original) A method in accordance with Claim 1 wherein effectuating all of the changes comprises changing the values of the plurality of tunable parameters in a predefined order.

5. (Previously Presented) A method in accordance with Claim 1 wherein effectuating all of the changes comprises either rebooting the operating system kernel before effecting said changes or effecting said change prior to rebooting the operating system kernel, conditioned upon whether at least one of the tunable parameters is a static tunable parameter.

6. (Original) A method in accordance with Claim 1 further comprising specifying a time at which said proposed changes are to be effectuated, and effectuating said proposed changes at the specified time.

7. (Original) A method in accordance with Claim 1 further comprising specifying an order in which said proposed new values are to take effect, and effectuating the changes in the specified order.

8. (Currently Amended) A computing apparatus having a central processing unit, a memory and a kernel, said computing apparatus configured to:

receive, from a user of the kernel, proposed changes to a plurality of tunable parameters of tunable kernel modules;

automatically determine whether application of the changes would violate a set of constraints, the determination including operability to automatically evaluate one or more alternate orders by which the proposed changes can be effectuated to identify whether a valid order exists that would not violate the set of constraints; and

effectuate either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively, the effectuation including operability to use the valid order to effectuate all of the proposed changes if such is identified by the automatically evaluation.

9. (Previously Presented) A computing apparatus having a central processing unit and a memory, said computing apparatus configured to:

specify proposed changes to a plurality of tunable parameters of tunable kernel modules;

determine whether application of the changes would violate a set of constraints;

effectuate either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively;

boot an operating system kernel having a tunable parameter registration function;

install a plurality of kernel modules, wherein at least some of said kernel modules are said tunable kernel modules having at least one said tunable parameter and at least one handler function, said handler function configured to at least one of obtain or change a value of a registered tunable parameter, wherein installing said tunable kernel modules includes writing information about its at least one tunable parameter into a registry memory maintained by the operating system kernel and indicating that the at least one tunable parameter is available for at least one of read or write access; and

use the tunable parameter registration function and the registry memory, registering the at least one tunable parameters of more than one kernel module;

and further wherein to specify proposed changes to a plurality of tunable parameters, said apparatus is configured to invoke one of said handler functions with a set of proposed new values for a corresponding selected subset of the registered tunable parameters;

wherein to determine whether application of the changes would violate a set of constraints, said apparatus is configured to determine whether the proposed new values are valid for every tunable parameter in the selected subset of the registered tunable parameters; and

wherein to effectuating either none or all of the changes, said apparatus is configured to change values of either every one or none of the selected subset of registered tunable parameters to the corresponding new values, conditioned upon a result of said determination of whether the proposed new values are valid.

10. (Original) An apparatus in accordance with Claim 8 wherein to effectuate all of the changes, said apparatus is configured to change the values of the plurality of tunable parameters in a predefined order.

11. (Original) An apparatus in accordance with Claim 8 wherein to effectuate all of the changes, said apparatus is further configured to either reboot the operating system kernel before effecting said changes or effect said change prior to rebooting the operating system kernel, conditioned upon whether at one of the tunable parameters is a static tunable parameter.

12. (Original) An apparatus in accordance with Claim 8 further configured to specify a time at which said proposed changes are to be effectuated, and to effectuate said proposed changes at the specified time.

13. (Original) An apparatus in accordance with Claim 8 further configured to specify an order in which said proposed new values are to take effect, and to effectuate the changes in the specified order.

14. (Currently Amended) A medium or media having machine-readable instructions recorded thereon configured to instruct a computing apparatus having a central processing unit, a memory and a kernel to:

receive, from a user of the kernel, proposed changes to a plurality of tunable parameters of tunable kernel modules;

automatically determine whether application of the changes would violate a set of constraints, instructions for the determination including instructions to automatically evaluate one or more alternate orders by which the proposed changes can be effectuated to identify whether a valid order exists that would not violate the set of constraints; and

effectuate either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively, instructions for the effectuation including instructions to use the valid order to effectuate all of the proposed changes if such is identified by the automatically evaluation.

15. (Previously Presented) A medium or media having machine-readable instructions recorded thereon configured to instruct a computing apparatus having a central processing unit and a memory to:

specify proposed changes to a plurality of tunable parameters of tunable kernel modules;

determine whether application of the changes would violate a set of constraints;

effectuate either none or all of the changes depending upon whether the changes were determined to violate any or none of the constraints, respectively;

load a tunable parameter registration function of a bootable operating system kernel;

install a plurality of kernel modules, wherein at least some of said kernel modules are said tunable kernel modules having at least one said tunable parameter and at least one handler function, said handler function configured to at least one of obtain or change a value of a registered tunable parameter, wherein installing said tunable kernel modules includes writing information about its at least one tunable parameter into a registry memory maintained by the operating system kernel and indicating that the at least one tunable parameter is available for at least one of read or write access; and

use the tunable parameter registration function and the registry memory, registering the at least one tunable parameters of more than one kernel module;

and further wherein to specify proposed changes to a plurality of tunable parameters, said medium or media also includes machine-readable instructions configured to instruct the computing apparatus to invoke one of said handler functions with a set of proposed new values for a corresponding selected subset of the registered tunable parameters;

wherein to determine whether application of the changes would violate a set of constraints, said medium or media also includes machine-readable instructions configured to instruct the computing apparatus to determine whether the proposed new values are valid for every tunable parameter in the selected subset of the registered tunable parameters; and

wherein to effectuating either none or all of the changes, said medium or media also includes machine-readable instructions configured to instruct the computing apparatus to change values of either every one or none of the selected subset of registered tunable parameters to the corresponding new values, conditioned upon a result of said determination of whether the proposed new values are valid.

16. (Original) A medium or media in accordance with Claim 15 wherein one of the tunable kernel modules is a dynamically loadable tunable module (DLKM), and further wherein said registering the at least one tunable parameter of the DLKM is performed the first time the DLKM is loaded; said

medium or media also having machine-readable instructions configured to instruct the computing apparatus to:

remove the DLKM from memory;

maintain the registration of said at least one tunable parameter of the DLKM while the DLKM is no longer in memory;

load the DLKM into memory at least a second time; and

access the value of at least one said tunable parameter of the DLKM using at least one said handler function, after the DLKM is loaded into memory at least a second time.

17. (Original) A medium or media in accordance with Claim 14 wherein to effectuate all of the changes, said medium or media also includes machine-readable instructions configured to instruct the computing apparatus to change the values of the plurality of tunable parameters in a predefined order.

18. (Original) A medium or media in accordance with Claim 14 wherein to effectuate all of the changes, said medium or media also includes machine-readable instructions configured to instruct the computing apparatus to either reboot the operating system kernel before effecting said changes or effect said change prior to rebooting the operating system kernel, conditioned upon whether at one of the tunable parameters is a static tunable parameter.

19. (Original) A medium or media in accordance with Claim 14 further having machine-readable instructions configured to instruct the computing apparatus to specify a time at which said proposed changes are to be effectuated, and to effectuate said proposed changes at the specified time.



20. (Original) A medium or media in accordance with Claim 14 further having machine-readable instructions configured to instruct the computing apparatus to specify an order in which said proposed new values are to take effect, and to effectuate the changes in the specified order.

21. (Cancel)

22. (Cancel)

23. (Cancel)

> remainder of page intentionally left blank <